```
##-------------------------------------------------

## R code for POLI502 Lab, week 5: Individual Exercises

## Written by: Howard Liu

##-------------------------------------------------

# 1. Exploring a data set ---------------------------

# We have learned several functions to explore a data set, including

dim(world.data)

head(world.data)

tail(world.data)

# There are some other functions we can use. For example, the names

# function tells us the names of all the variables included in a data frame

# object.

names(world.data)

# The colnames function gives us the same results as well.

colnames(world.data)
```

We can also apply the summary function without specifying variable

names. Then, R will provide the summary of ALL the variables in cluded

in a data frame object.

summary(world.data)

The str() function tells us the structure of a data frame object,

meaning that it tells us which variables are factor, which ones are

numerical, which ones are logical, etc.

str(world.data)

#clean up my workplace

```
rm(list=ls(all=TRUE))
cat("\014")
```

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(dplyr)
```

#upload data from computer

```
world.data = read.csv("~/Desktop/2025_Fall/Poli_502/Poli502_Feng/Data/world.csv")
```

2. Summarizing categorical variables

The output from the str function above tells us that there are many factor

variables in the data set. For example, the democ_regime variable

is a factor variable (nominal-level). Summarize the information contained

in this variable by creating a frequency table.

frequency table for democratic regime variable

```
table(world.data $ democ_regime)
```

```
No Yes
75 114
```

The typerel variable is another factor variable.

This variable measures predominant religion in a given country.

Create a frequency table for this variable.

frequency table for religion variable

```
table(world.data $ typerel)
```

```
       eastern          Hindu         Jewish         Muslim       Orthodox 
            15              2              1             50             13 
         other     Protestant Roman Catholic 
            12             35             63 
```

Make this frequency table vertical using the data.frame function

convert frequency table into vertical format

```
data.frame( table(world.data $ typerel) )
```

```
           Var1 Freq
1        eastern   15
2          Hindu    2
3         Jewish    1
4         Muslim   50
5       Orthodox   13
6          other   12
7     Protestant   35
8 Roman Catholic   63
```

We have seen in the lecture that we often report RELATIVE frequencies

as well as raw frequencies. Relative frequencies can be obtained by

dividing each of the raw frequency values by the total number of

observations. Let's see how we do this.

To do so, it is better if we create a new object that stores the

frequency table. Let's create an object called ft.colony that is

equal to the vertical frequency table for the colony variable,

as follows.

frequency table for colony variable

```
ft.colony <- data.frame( table(world.data $ colony) )
```

To make sure we did this correctly, let's take a look

```
ft.colony # view the table
```

```
           Var1 Freq
1       Belgium    3
2        France   28
3   Netherlands    4
4          none   20
5         Other   15
6       Ottoman    2
7      Portugal    8
8  Soviet Union   27
9         Spain   21
10           UK   63
```

We can see that the first column, Var1, records all possible

values and the second column, Freq, records the raw frequency.

To convert the raw frequencies into relative frequencies, we divide

the values by the sum of Freq.

As we learned before, we use the sum function to calculate the sum of

all the values, as follows.

```
sum( ft.colony $ Freq ) # calculate total count of countries
```

```
[1] 191
```

The relative frequencies are Freq divided by sum(ft.colony $ Freq)

```
ft.colony $ Freq / sum( ft.colony $ Freq )# convert raw frequencies to relative freque
```

```
 [1] 0.01570681 0.14659686 0.02094241 0.10471204 0.07853403 0.01047120
 [7] 0.04188482 0.14136126 0.10994764 0.32984293
```

Alternatively, we can use the prop.table function to obtain the

same results

```
prop.table(ft.colony $ Freq) # alternative way to get proportions
```

```
 [1] 0.01570681 0.14659686 0.02094241 0.10471204 0.07853403 0.01047120
 [7] 0.04188482 0.14136126 0.10994764 0.32984293
```

We would want to convert these further into percentages.

To make a ratio into a percentage, we simply multiply it by 100

```
prop.table(ft.colony $ Freq) * 100 # convert proportions into percentages
```

```
 [1]  1.570681 14.659686  2.094241 10.471204  7.853403  1.047120  4.188482
 [8] 14.136126 10.994764 32.984293
```

We would want to round these numbers to simplify the representation.

As we leanred two weeks ago, we use the round function to do that.

```
round(prop.table(ft.colony $ Freq) * 100, digits = 2) # round percentages to 2 decimal
```

```
 [1]  1.57 14.66  2.09 10.47  7.85  1.05  4.19 14.14 10.99 32.98
```

Finally, we want to insert these numbers into the frequency table

we created and stored in ft.colony.

```
ft.colony # create frequency table again
```

```
           Var1 Freq
1       Belgium    3
2        France   28
3   Netherlands    4
4          none   20
5         Other   15
6       Ottoman    2
7      Portugal    8
8  Soviet Union   27
9         Spain   21
10           UK   63
```

How do we do it?

We do this by creating a new column in the ft.colony object.

```
# As we learned last week, we use the $ symbol to create a new column

# in a data frame object, as follows

ft.colony $ Percent <- round(prop.table(ft.colony $ Freq) * 100, digits = 2)

# Now, our frequency table contains three columns, as follows

ft.colony

# Finally, we may want to change the column name for the first column

# from “Var1” to something more intuitive.

# To do so, we use the colnames function, as follows

colnames(ft.colony)[colnames(ft.colony) == “Var1”] <- “Colonizer”

ft.colony

# We can see that about 33% of the countries in the world are former colonies

# of the UK, about 15% of them are former colonies of France, about 10% of them

# were never colonized, etc.

# To summarize the steps to create a frequency table:
```

```
ft.colony <- data.frame( table(world.data $ colony) ) # create frequency table again
ft.colony $ Percent <- round(prop.table(ft.colony $ Freq) * 100, digits = 2) # add per
colnames(ft.colony)[colnames(ft.colony) == "Var1"] <- "Colonizer" # rename first colum
ft.colony
```

```
      Colonizer Freq Percent
1       Belgium    3    1.57
2        France   28   14.66
3   Netherlands    4    2.09
4          none   20   10.47
5         Other   15    7.85
6       Ottoman    2    1.05
7      Portugal    8    4.19
8  Soviet Union   27   14.14
9         Spain   21   10.99
10           UK   63   32.98
```

Create a frequency table for the typerel variable

```
data.frame(table(world.data $ typerel)) # frequency table for typerel variable
```

```
           Var1 Freq
1       eastern   15
2         Hindu    2
3        Jewish    1
4        Muslim   50
5      Orthodox   13
6         other   12
7    Protestant   35
8 Roman Catholic   63
```

Which religion is the most “popular” in the world?

That is, what is the mode of the “typerel” variable?

Roman Catholic is the most popular in the world. # # # # What is the percentage of countries where muslim is the majority? # 26.18% # # # # Create a frequency table for democ_regime #

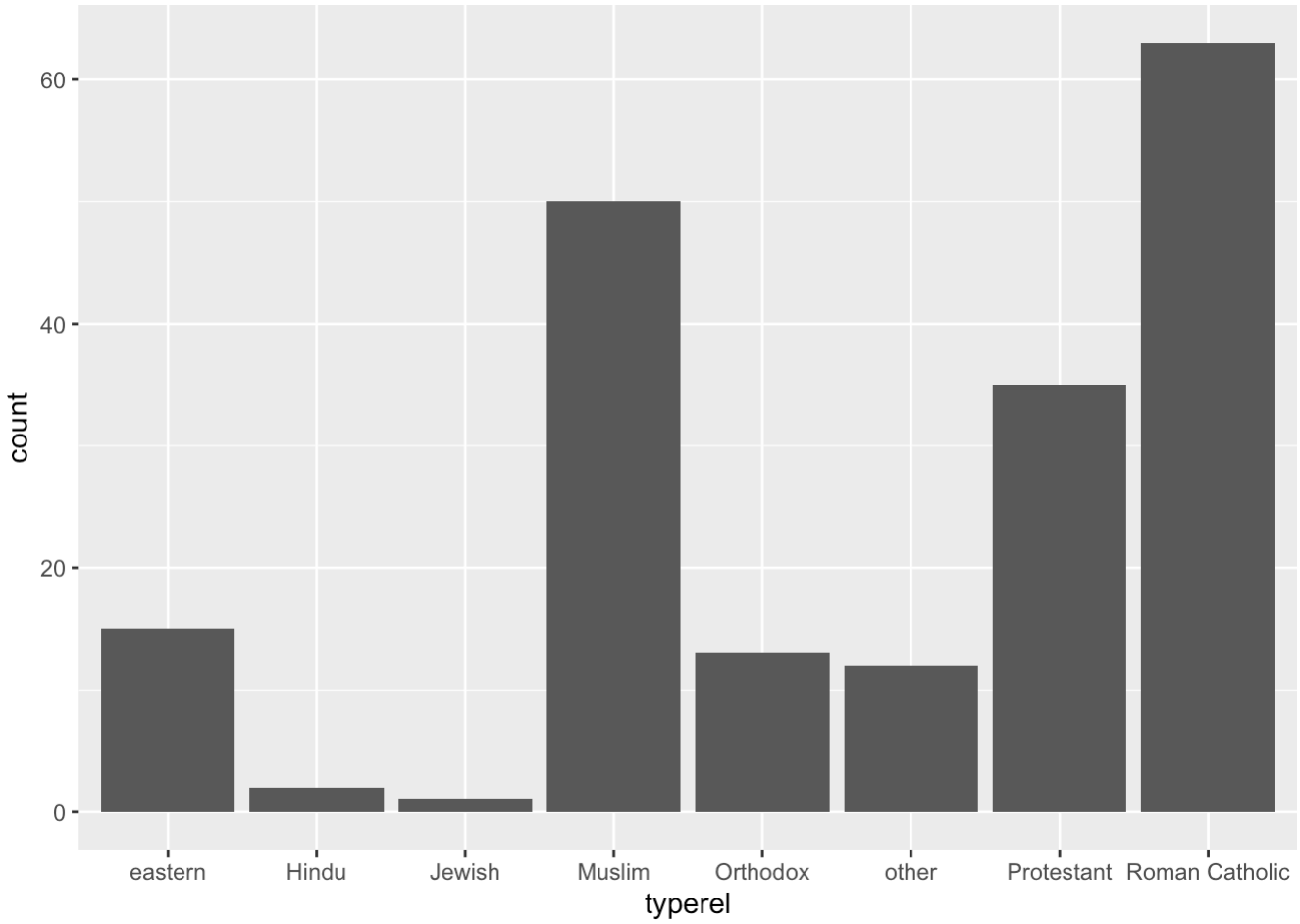
```
data.frame(table(world.data $ democ_regime))# frequency table for democratic regime va
```

```
  Var1 Freq
1   No   75
2  Yes  114
```

What percentage of countries have a democratic regime?

60.3% # # # # Create a bar chart to summarize the typerel variable #

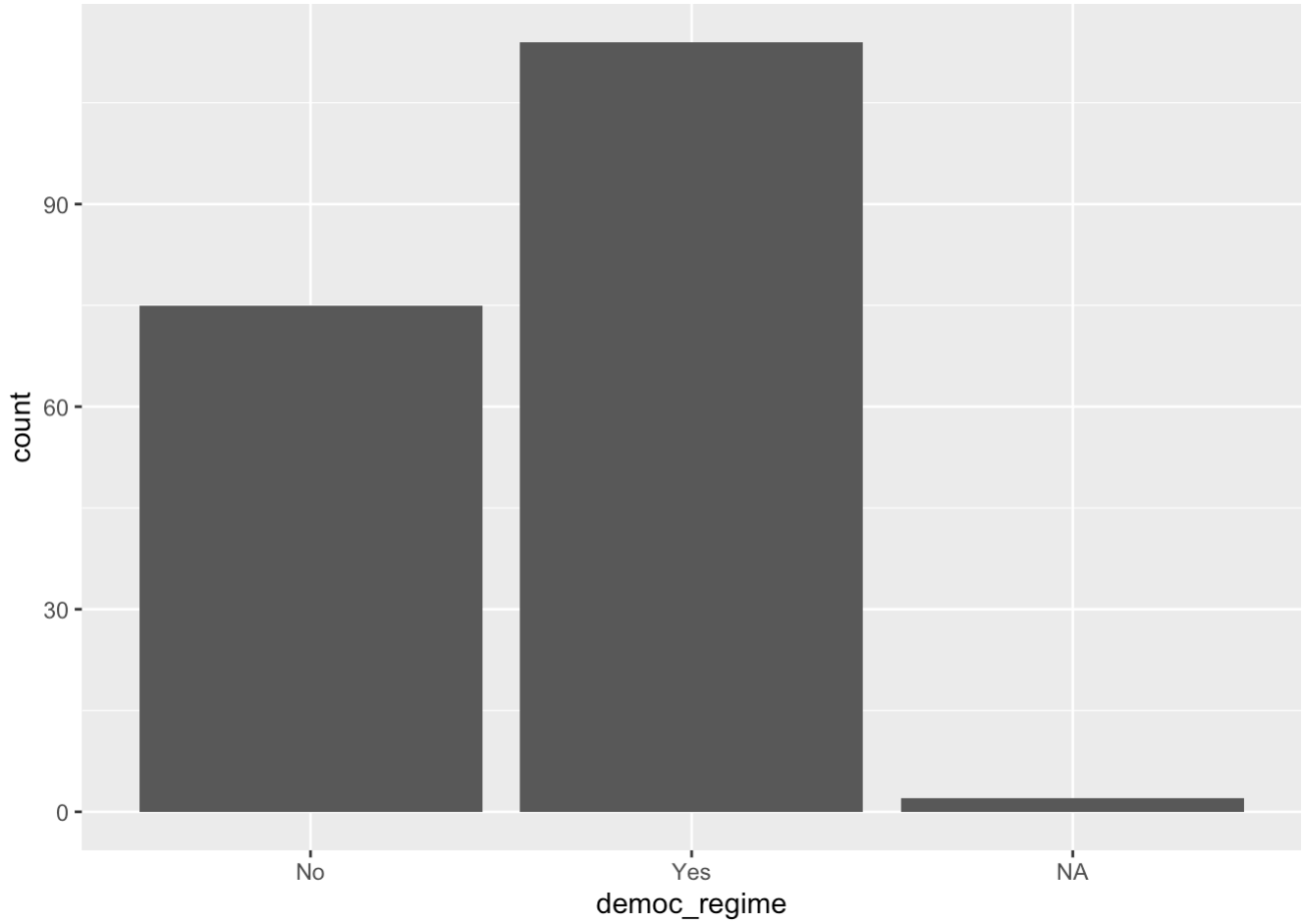
```
ggplot(world.data, aes(x = typerel)) +
  geom_bar()
```




```
# bar chart for typerel with label^

#
#
# # Create a bar chart to summarize the democ_regime variable
#

ggplot(world.data, aes(x = democ_regime)) +
  geom_bar() # bar chart for democratic regime variable
```



3. Making ggplot graphs look nicer

We have seen how to create a graph using the ggplot function

ggplot(world.data, aes(x = colony)) + geom_bar()

The command above is the easiest way to produce a simple ggplot

graph, but we would want to modify some parts of the graph, such as

axis labels. For example, the graph above currently says

“colony” on the x-axis and “count” on the y-axis. We may want to

modify them so they can be more informative.

When we want to modify graphs, we usually create a ggplot graph

and store it into an object. Then we gradually add some features

to modify them. The above command can be re-written as follows:

g <- ggplot(world.data) # Tells R which data frame contains the variable to plot

g <- g + aes(x = colony) # Tells R which variable to plot

g <- g + geom_bar() # Tells R what type of graph we want

g # Tells R to show the contents of the object g

Now that we stored the graph into an object called g, we can

modify graph appearances by adding more options.

To change the label for the x-axis, we use the xlab option, as follows

g <- g + xlab(“Colony of Which Country?”)

g

```
# Similarly, we can modify the label for the y-axis

g <- g + ylab("Number of countries")

g

# If you want to change the text size for axes, do

g <- g + theme(axis.text.x = element_text(size = 12))

g

# We can save this graph as a PDF file using the ggsave function.

ggsave(file = "colony_bar.pdf", width = 10, height = 8)

# The file option specifies the file name of the PDF file

# you want to create.

# The width and height option control the width and height of the

# PDF file, respectively.

# Once you save a graph in a PDF, you can easily embed it in a

# Word document simply by drag & drop.

# To summarize what we have done so far,

g <- ggplot(world.data)
```

g <- g + aes(x = colony)

g <- g + geom_bar()

g <- g + xlab("Colony of Which Country?")

g <- g + ylab("Number of countries")

g <- g + theme(axis.text.x = element_text(size = 12))

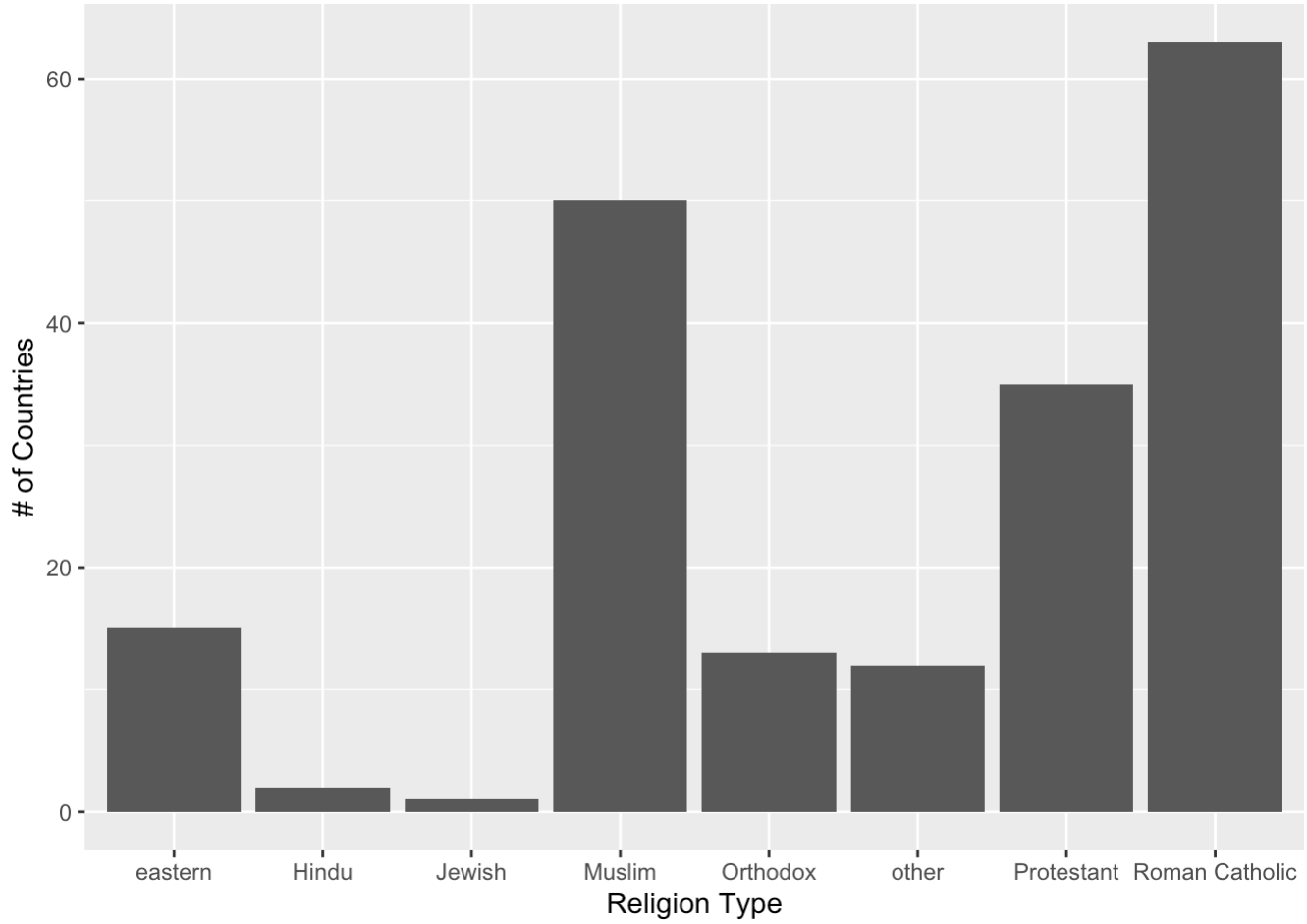
g

ggsave(file = "colony_bar.pdf", width = 10, height = 8)

Create a bar chart for the typerel variable, and save it as a PDF file

WRITE YOUR COMMANDS HERE

```
ggplot(world.data, aes(x = typerel)) +
  geom_bar() +
  labs(x = "Religion Type", y = "# of Countries")
```




```
ggsave("typerel1_nar.pdf") # save the typerel bar chart as PDF
```

```
Saving 7 x 5 in image
```

4. Summarizing numerical variables

There are two variables in the data set, gini04 and gini08, that measure

the levels of economic inequality in a country numerically.

These are what's called Gini coefficient (Gini index or Gini ratio), which

takes values between 0 and 1 (or 0% and 100%). A value of 0 corresponds to

the “perfect equality” case, where everyone in a country is earning the same

amount of money, whereas a value of 1 (100%) corresponds to the maximal

inequality case, where one person is earning ALL the money in a country and

everyone else is earning nothing. The gini04 variable is from the year 2004

whereas the gini08 variable is from the year 2008.

Numerically summarize the gini04 variable.

That is, calculate and present the measures for central tendency and

those for dispersion.

WRITE YOUR COMMANDS HERE

```
summary(world.data$gini04)  # numerical summary of gini04 variable
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  24.40   32.42   37.95   40.14   46.88   70.70      65 
```

```
sd(world.data$gini04, na.rm = TRUE) # standard deviation of gini04 variable
```

```
[1] 10.35998
```

Numerically summarize the gini08 variable.

That is, calculate and present the measures for central tendency and

those for dispersion.

WRITE YOUR COMMANDS HERE

```
summary(world.data$gini08) # numerical summary of gini08 variable
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  24.70   33.55   39.20   40.74   47.10   74.30      64 
```

```
sd(world.data$gini08, na.rm = TRUE)# standard deviation of gini08 variable
```

```
[1] 9.999242
```

Compare the distributions of gini04 and gini08. Do you think that the

level of economic inequality is getting worse, getting better, or neither?

Why or why not?

Note: I'm not asking why it is getting better, worse, or neither;

I'm asking on what basis you can conclude that it's getting better or worse.

WRITE YOUR ANSWER HERE

They look very similar in regard to mean median and dispersion. Neither show a fundemental shift between 2004 and 2008 so it looks like inequality is stagnant (neither getting better or worse). Both statsistics do not show any big changes across the years measured.

Create a histogram of gini04

Modify the axis labels accordingly to make them informative and intuitive.

Save the graph as a PDF file.

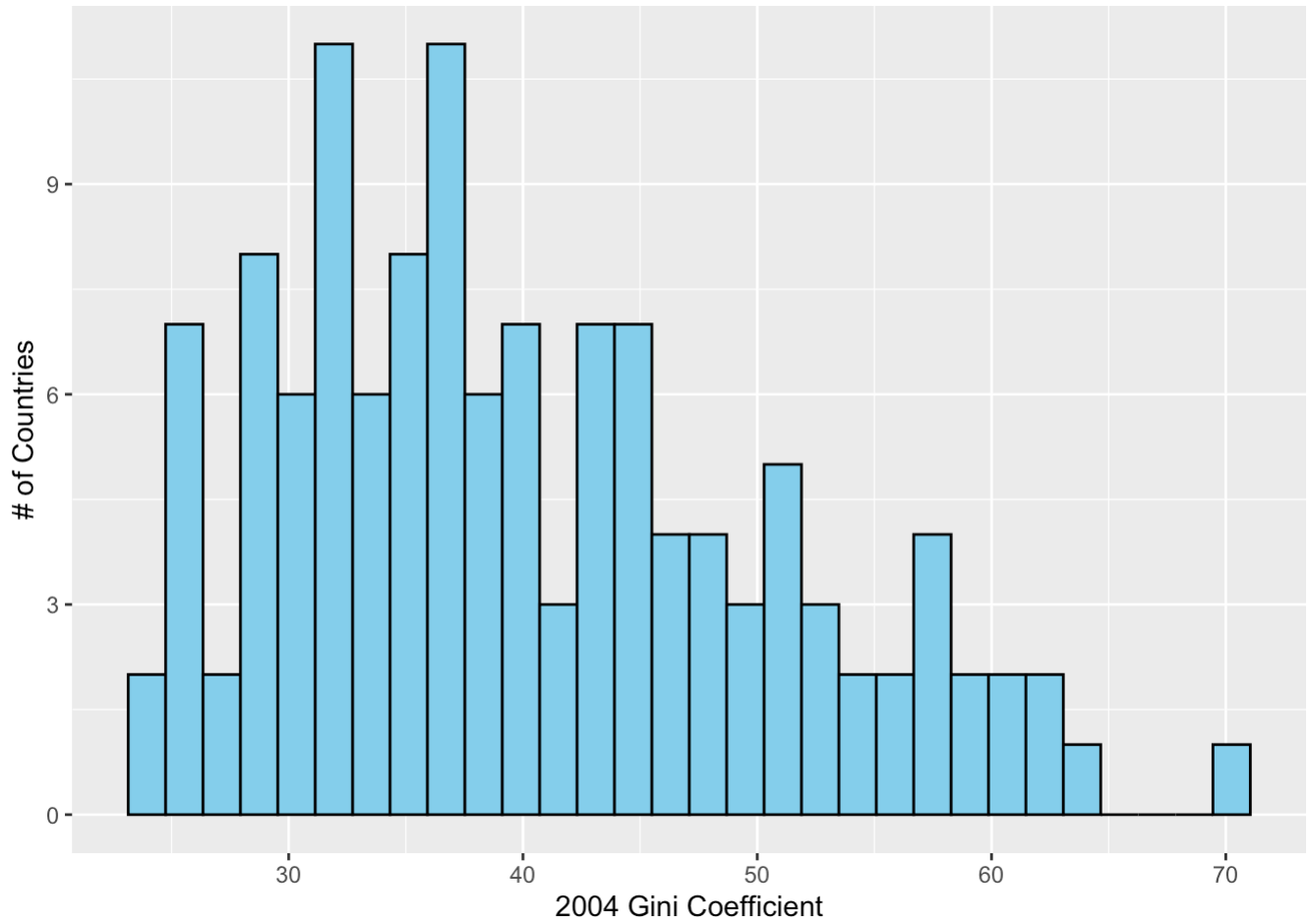
WRITE YOUR COMMANDS HERE

histogram for gini04 variable

```
ggplot(world.data, aes(x = gini04)) +
  geom_histogram(fill = "skyblue", color = "black") +
  labs(x = "2004 Gini Coefficient", y = "# of Countries")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 65 rows containing non-finite outside the scale range
(`stat_bin()`).
```




```
ggsave("gini04_hist.pdf")
```

```
Saving 7 x 5 in image
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
Warning: Removed 65 rows containing non-finite outside the scale range
(`stat_bin()`).
```

Create a histogram of gini08

Modify the axis labels accordingly to make them informative and intuitive.

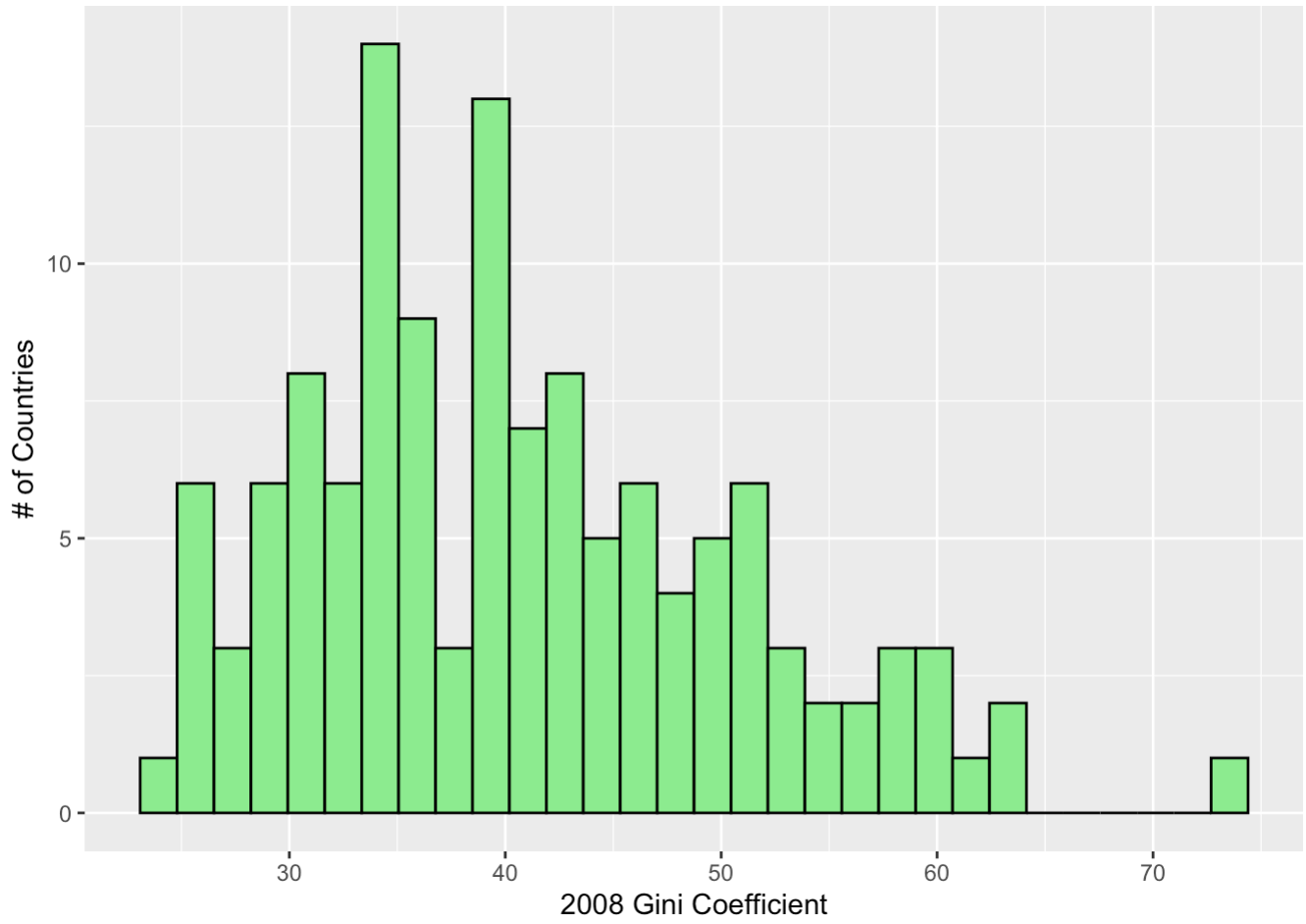
Save the graph as a PDF file.

histogram for gini08 variable

```
ggplot(world.data, aes(x = gini08)) +
  geom_histogram( fill = "lightgreen", color = "black") +
  labs(x = "2008 Gini Coefficient", y = "# of Countries")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 64 rows containing non-finite outside the scale range
(`stat_bin()`).
```



```
ggsave("gini08_hist.pdf", width = 10, height = 8)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 64 rows containing non-finite outside the scale range
(`stat_bin()`).
```

Compare the distributions of gini04 and gini08 graphically by

placing the two PDF files you just created side by side.

Do you confirm the conclusion you derived previously?

They show extremely similar shapes and values. The graphs confirm the previous conlcusion in that the overal level of economic inequality stayed the same from 2004 to 2008.

As we saw in the lecture, we sometimes create histograms for different

values of a nominal-level variable. For example, we may want to create

separate histograms of gini04 for countries in different regions.

To do so, we use the facet_wrap option, as follows.

COPY & PASTE THE CODE YOU WROTE TO PRODUCE HISTOGRAM FOR gini04 HERE

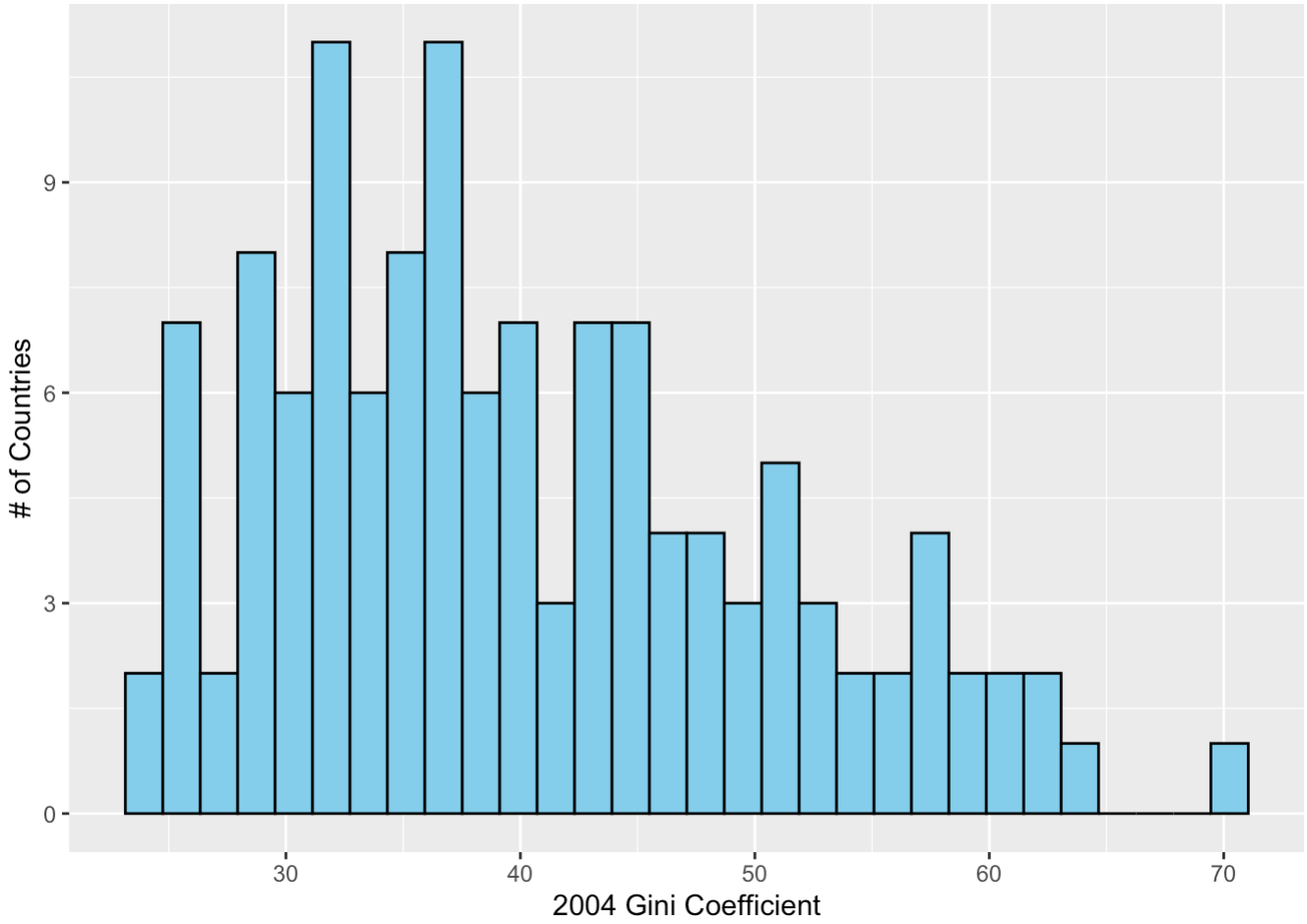
histogram of gini04 with labels

```
ggplot(world.data, aes(x = gini04)) +
  geom_histogram( fill = "skyblue", color = "black") +
```

```
  labs(x = "2004 Gini Coefficient", y = "# of Countries")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 65 rows containing non-finite outside the scale range
(`stat_bin()`).
```




```
ggsave("gini04_hist.pdf")
```

```
Saving 7 x 5 in image
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 65 rows containing non-finite outside the scale range
(`stat_bin()`).
```

g <- g + facet_wrap(~ region)

g

We can see that Scandinavian countries and Western European countries

have, on average, lower Gini coefficients (= they are more egalitarian),

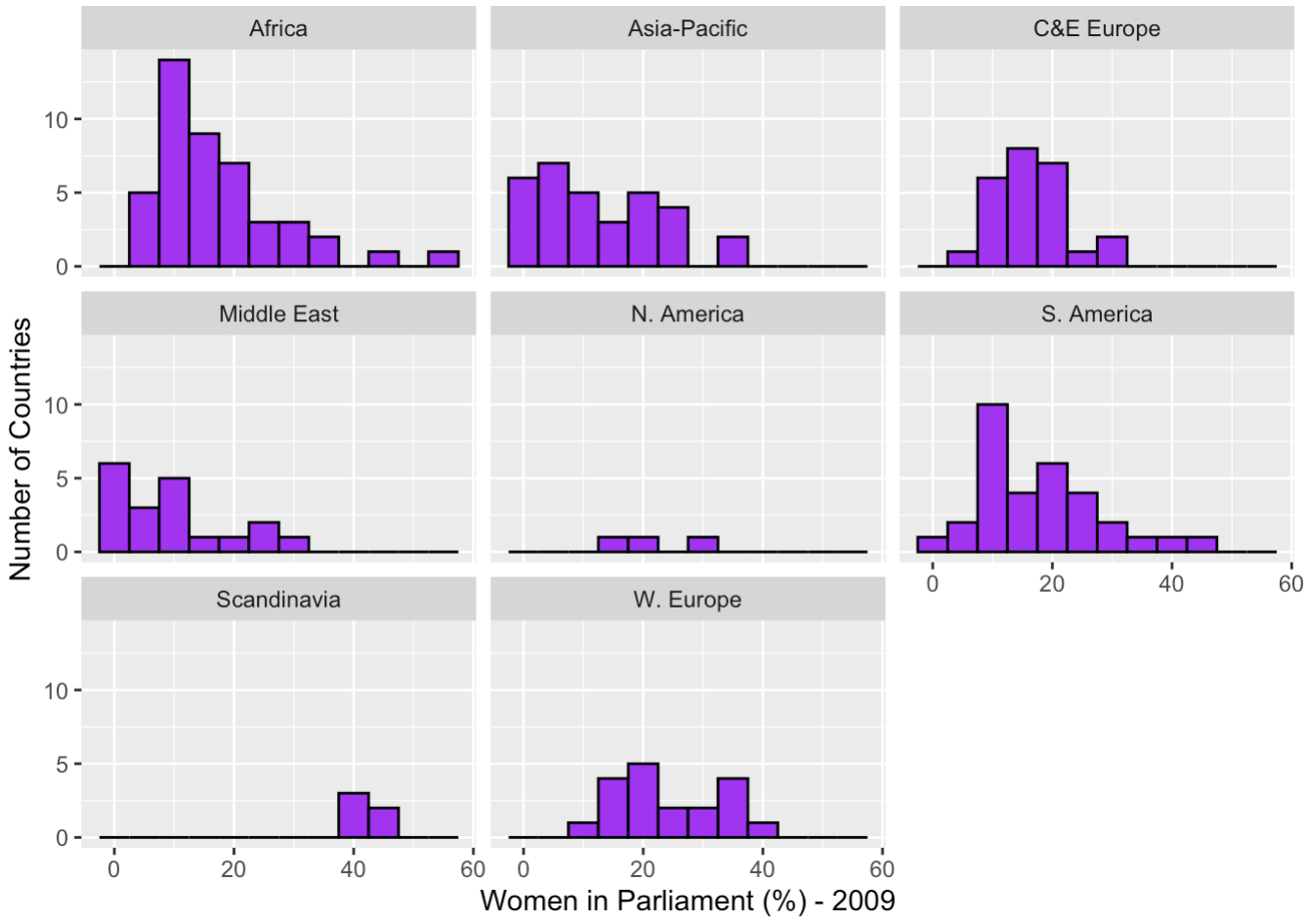
whereas countries in South America have relatively high values.

Create separate histograms of women09 for countries in different regions.

histogram of women09 variable

```
g <- ggplot(world.data, aes(x = women09)) +
  geom_histogram(binwidth = 5, fill = "purple", color = "black") +
  labs(x = "Women in Parliament (%) - 2009", y = "Number of Countries")
g <- g + facet_wrap(~ region)
g
```

```
Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_bin()`).
```



We may want to do the same using numerical methods.

That is, we may want to obtain central tendencies and dispersions for

a numerical variable for different groups.

To do so, we use the by function.

The by function take the following form

by(VARIABLE_YOU_WANT_TO_ANALYZE, GROUP, FUNCTION)

That is, you provide

(1) an interval-level variable you want to summarize first,

(2) a comma

(3) a nominal variable that separates observations into groups

(4) a comma

(5) a function you want to apply (such as summary, mean, median, sd, etc.)

For example, to obtain numerical summaries of gini04 for different regions,

we write

by(world.data $ gini04, world.data $ region, summary)

Calculate the standard deviation of gini04 for different regions using

the by function

Hint: we still need to take care of the missing value problem.

Use the na.rm = TRUE option.

WRITE YOUR COMMAND HERE

standard deviation of gini04 by region

```
by(world.data$gini04, world.data$region, sd, na.rm = TRUE)
```

```
world.data$region: Africa
[1] 11.06417
------------------------------------------------------------
world.data$region: Asia-Pacific
[1] 6.651417
------------------------------------------------------------
world.data$region: C&E Europe
[1] 5.167651
------------------------------------------------------------
world.data$region: Middle East
[1] 3.314901
------------------------------------------------------------
world.data$region: N. America
[1] 10.89327
------------------------------------------------------------
world.data$region: S. America
[1] 6.329504
------------------------------------------------------------
world.data$region: Scandinavia
[1] 0.9831921
------------------------------------------------------------
world.data$region: W. Europe
[1] 3.679761
```

Which region has the smallest dispersion?

WRITE YOUR ANSWER HERE

The region with the smallest dispersion of gini04 is Western Europe because its countries are more similar to each other with regard to income inequality.

End of file